# OBJECTIVE:

First, we will apply a logistic model by taking features of URLs with an accuracy

Our Main objective is to rank urls on the basis of the features (which we will discuss further), with the help of this rank we will be able to find the top n urls

And the ranking will be on the basis of how much the urls is malicious, more the probability that the urls is malicious more is its ranking

First find the nature of urls by logistic regression and after that we will check it’s ranking by Bayes’ theorem

# CONCEPT:

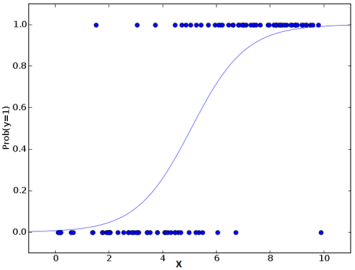
(With the help of Bayes’ criterion we are able to find weights of each feature of URLs , with the weights we will combine features value and weight in such a way according to the data (types of data for example Ordered data ,categorical data ,nominal data etc. ) that it will give us a combined result which will become or data on which we will set our threshold and we will verify our logistic model it’s accuracy )

# ABSTRACT: -

**What is a Classification Problem?** We identify problem as classification problem when independent variables are continuous in nature and dependent variable is in categorical form i.e. in classes like positive class and negative class. The real-life example of classification example would be, to categorize the mail as spam or not spam, to categorize the tumor as malignant or benign and to categorize the transaction as fraudulent or genuine. All these problem’s answers are in categorical form i.e. Yes or No

Logistic Regression is one of the basic and popular algorithms to solve a classification problem. Logistic Regression, is a mathematical model used in statistics to estimate (guess) the probability of an event occurring having been given some previous data. Logistic Regression works with binary data, where either the event happens (1) or the event does not happen (0). In our case as well the data is benign or malicious. So, given some feature x it tries to find out whether some event y happens or not. So, y can either be 0 or 1. In the case where the event happens, y is given the value 1, If the event does not happen, then y is given the value of 0. Logistic Regression uses Sigmoid function. An explanation of logistic regression can begin with an explanation of the standard logistic function. The logistic function is a Sigmoid function, which takes any real value between zero and one

Equation: - sigmoid function = s(t)= (e^t)/((e^t) +1)



The above figure shows the fitting of model on binary data taking two values 0 and 1

# STATISTICAL CONCEPT USED: -

# BAYES THEOREM: describes the probability of an event based on prior knowledge of conditions that might be related to the event.

P(A/B) =probability of A event when we are given event B

Hence, it is a conditional probability

FORMULA OF BAYES THEOREM: -

P(B/A) =(P(A/B) \*P(B)) / (P(A))

Where,

According to our use we had taken:

B=set of malicious urls

P(B) = Probability of feature F in malicious urls

A = set of urls with feature F in legitimate dataset which can say that urls is malicious

P(A)=probability of feature F in legitimate dataset

P(A/B) = probability of urls in dataset when we have given that the particular feature is present in the urls

P(B/A) =probability of malicious urls in dataset

# MATHEMATICAL CALCULATIONS AS AN EXAMPLE: -

dataset, 1000 phishing mails and 1000 legitimate mails

**IP address More Dots Encoded Suspicious**

**Symbol Characters**

**Phishing mail** 40 60 10 10

**Legitimate mail** 10 10 0 0

**Feature F1 (Lexical features)**

The feature F1 involves the occurrence of lexical features that

appeared in 120 phishing URLSs and 20 legitimate URLSs. Hence its

probability is calculated as follows.

P(B=1|A) =P(A|B=1)/(P(A|B=0) + P(A|B=1))

P(B|A) =(120|1000)/((120/1000)+(20/1000))

P(B/A)= 0.86,

P(B(Phishing)) = P(B’(Legitimate)) = 0.5.

Now , 0.86 is the weight of feature f1

Similarly, we can find weights of different features

# FEATURES: - (for urls)

* urls length =length of urls
* hostlength = length of host(for example, http://www.google.comsg/webhp?hl=zh-CN , length=19)
* pathlength=length of path (for example, http://www.google.comsg/webhp?hl=zh-CN,length=14)
* fd\_ length= length of fetch domain
* tldlength= length of top level domain
* 'count-'=number of “-“ in urls
* 'count@'=number of “@“ in urls
* 'count?'=number of “?“ in urls
* 'count%'=number of “%“ in urls
* 'count.'=number of “.“ in urls
* 'count=’ = number of “=“ in urls
* 'count-http' = number of “http“ in urls
* 'count- HTTPs’=number of “https“ in urls
* 'count-www'= number of “www“ in urls
* 'count-digits'=number of “digits“ in urls
* 'count-letters'=number of “alphabet“ in urls
* 'count dir'=directory is present or not
* 'use of ip'=ip address is present or not (for example http://52.41.234.24/applcation/login.aspx hence ip is present in this urls )

Important feature: -

Page rank represents the relative importance of a page within a set

of web pages. The higher the page rank, the more important is the page.

Phishing web pages are short lived and thus either have a very low page rank

or their page rank does not exist. Page rank is a link analysis algorithm first

used by Google, in which each document on the web is assigned a numerical

weight from 0 to 10, with 0 indicating least popular and 10 meaning most

popular. A score value of *􀃭*1 is assigned when the page rank value for a

particular webpage is not available. **Page rank (F3)** Page rank represents

the relative importance of a page within a set of web pages. The higher the

page rank, the more important is the page. Phishing web pages are short

lived and thus either have a very low page rank or their page rank does not

exist. Page rank is a link analysis algorithm first used by Google, in which

each document on the web is assigned a numerical weight from 0 to 10, with

0 indicating least popular and 10 meaning most popular. A score value of *􀃭*1

is assigned when the page rank value for a particular webpage is not available.

# INTRODUCTION AND POC (PROOF OF THE CONCEPT)

Introduction: -

There are two ways to test a model, static testing and dynamic testing. Main Difference between Static testing and Dynamic Testing. Static testing is done before the code deployment whereas dynamic testing is after the code deployment. Static testing is done in verification stage whereas dynamic testing is done in validation stage.

In our case we are doing static testing because we are verifying our model by statistical technique hence, we are verifying our model

POC: -

First we will take a set of urls and we will train and test our logistic model we created , with the help of these urls, we had trained this model to test whether the urls are benign or malicious , after that we will take a different set of urls and pass that urls through the model and it will give us whether the new urls are benign or malicious as an result with an accuracy we had trained the model with.

Following is the data we had trained and test our model on

|  |  |  |  |
| --- | --- | --- | --- |
| Urls | label | result |  |
| <https://www.google.com> | benign | 0 |  |
| http://185.186.77.106/ntpd | malicious | 1 |  |
| https://www2.carinho-portal.com/ | benign | 0 |  |
| <https://www.baidu.com> | benign | 0 |  |
| <http://68.183.36.8/roose> | malicious | 1 |  |
| http://stadtmisr.com/f/zzp/zzp.exe | malicious | 1 |  |
| <http://68.183.36.8/grape> | malicious | 1 |  |
| <http://68.183.36.8/tuan> | malicious | 1 |  |
| http://kbc-assist.be/ | malicious | 1 |  |
| http://www.ihgcc.com/ | malicious | 1 |  |

|  |
| --- |
| urls |
| https://www.klinikjejaringpadjadjaran.com/onedrivv... | | | | | |
| http://verdilugano.ch/cn/LinkedIn.html | | | |
| https://airbit.icu/sec/ok/index.php | | | |
| https://adalenx.top/find/proposal/v3/v3/index1.htm... | | | | | |
| https://tntadventures.info/ms/login/ | | | |
| https://frederiksbergxc.top/admin/procurement/page/index1.html?n6tVpBN... | | | | | | |
| http://paypal.orturlss.com/b798da/en/season.php?country.x=2b78be1912568d473f | | | | | | |
| http://m.fb-com-rxfsbloqgh.rimass.ma/gateway.php | | | | | |
| http://service.progefai.it/web/whatsapp/index/en | | | | | |
| http://secure.bnz.co.nz.ieb4esuing.com/auth/personal-login | | | | | |
| http://m.fb-com-rxfsbloqgh.rimass.ma/gateway.php | | | | | |
| https://thenrsca.com/chasupdt2018/a4350073c346984b436c7ae7a9db501d | | | | | | |
| https://buff.ly/2Iqzz3P | | |
| https://rhodesomles.com/naso/ | | | |
| http://dycehong.duckdns.org/ | | |
| http://sicurezzabpol.com/jod-fcc/otp/step5.php?authToken=5a6d043dd2140... | | | | | | |
| http://nrscas.com/chasupdt2018/55cdcdbf93d4195a43c7c6e35cea88da/login.... | | | | | | |
| https://airbit.icu/eer/ok/index.php | | | |
| https://thenrsca.com/chasupdt2018 | | | |
| http://continence-foundation.com/assets/web | | | | |
| http://212.237.49.211/ | | |
| http://frisco.cc/ | |
| https://icloud.com.signin-support-id.live/signin... | | | | |
| https://bankofamerica.trust11.com/ | | | |
| http://app.smartsheet.com/b/home | | | |
| http://lu78.cc/ | |
| https://apple.com.support-login.us/signin | | | | |
| http://salon-jadore.ru/wp-content/plugins/js\_composer00000000000/vendo... | | | | | | |
| http://0794aa.com/ | |
| http://jppost-to.com:81/sa.html | | | |
| https://www.onehoursitefix.com/ | | | |
| <https://forums.runescape.com-f.ru> | | | |
| Above are the urls passed by the model and it gave the following result(0 represent benign and 1 represent malicious)  [ 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 ] | | | |

Now we will check the accuracy of the model by the statistical concept explained in abstract

Here , number of malicious and benign are same hence , P(B)=0.5 and P(B’)=0.5

Where ,P(B)=probability of malicious in dataset

P(B’)= probability of benign in dataset

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | host len | path len | fd len | tld len | count '\_' | count '@' | count '.' | count '=' |
| Beni-gn | 7 | 7 | 1 | 7 | 1 | 0 | 0 | 1 |
|  |  |  |  |  |  |  |  |  |
| Malic-ious | 6 | 9 | 5 | 5 | 2 | 0 | 5 | 0 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| http | https | www | Digit | Letter | Dir | Use of ip | Short url |
| 8 | 8 | 8 | 1 | 9 | 6 | 0 | 0 |
| 9 | 3 | 2 | 6 | 9 | 9 | 3 | 0 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **NOW WEIGHTS FOR BENIGN URLS’S**  F1=host­ length | |  |  |  |  |  |
| P(B/A)= (7/9)/(6/9+7/9)  P(B/A)=0.53845 | | | | | | |
| W1=WEIGHT OF HOST LENGTH IN B WHEN GIVEN BENIGN =0.53845  F2=past length  P(B/A)= (7/9)/(9/9+7/9)  P(B/A)=0.4375  W2=WEIGHT OF PATH LENGTH IN B WHEN GIVEN BENIGN =0.4375  F3=TLD length  P(B/A)= (7/9)/(5/9+7/9)  P(B/A)=0.5833  M3=WEIGHT OF TLD LENGTH IN B WHEN GIVEN BENIGN =0.5833  F4=FD length  P(B/A)= (1/9)/(5/9+1/9)  P(B/A)=0.1667  M4=WEIGHT OF FD LENGTH IN B WHEN GIVEN BENIGN =0.16  F5= COUNT ‘\_’  P(B/A)= (1/9)/(2/9+1/9)  P(B/A)=0.3334  M5=WEIGHT OF COUNT ‘\_’ IN B WHEN GIVEN BENIGN =0.3334  F6= COUNT ‘@’  P(B/A)= (0/9)/(0/9+0/9)  P(B/A)=0.0  M6=WEIGHT OF COUNT ‘@’ IN B WHEN GIVEN BENIGN =0.0  F7= COUNT ‘.’  P(B/A)= (0/9)/(5/9+0/9)  P(B/A)=0  M7=WEIGHT OF COUNT ‘.’ IN B WHEN GIVEN BENIGN =0.0  F8= COUNT ‘=’  P(B/A)= (1/9)/(0/9+1/9)  P(B/A)=1  M8=WEIGHT OF COUNT ‘=’ IN B WHEN GIVEN BENIGN =1  F9= HTTP  P(B/A)= (8/9)/(9/9+8/9)  P(B/A)=0.4706  M9=WEIGHT OF HTTP IN B WHEN GIVEN BENIGN =0.4706  F10= HTTPS  P(B/A)= (8/9)/(3/9+8/9)  P(B/A)=0.7273  M10=WEIGHT OF HTTPS IN B WHEN GIVEN BENIGN =0.7273  F11= WWW  P(B/A)= (8/9)/(2/9+8/9)  P(B/A)=0.8  M11=WEIGHT OF WWW IN B WHEN GIVEN BENIGN =0.8  F12= DIGITS  P(B/A)= (1/9)/(6/9+1/9)  P(B/A)=0.1428  M12=WEIGHT OF DIGITS IN B WHEN GIVEN BENIGN =0.1428  F13= LETTER  P(B/A)= (9/9)/(9/9+9/9)  P(B/A)=0.5  M13=WEIGHT OF LETTER IN B WHEN GIVEN BENIGN =0.5  F14= DIR  P(B/A)= (9/9)/(9/9+6/9)  P(B/A)=0.4  M14=WEIGHT OF DIR IN B WHEN GIVEN BENIGN =0.4  F15= ‘USE OF IP’  P(B/A)= (0/9)/(3/9+0/9)  P(B/A)=0.0  M15=WEIGHT OF ‘USE OF IP’ IN B WHEN GIVEN BENIGN =0.0  F16= SHORT URLS  P(B/A)= (0/9)/(0/9+0/9)  P(B/A)=0.0  M16=WEIGHT OF SHORT URLS IN B WHEN GIVEN BENIGN =0.0  TABLE FOR WEIGHT OF FEATURES   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | Host len | path len | fd len | tld len | count '\_' | count '@' | count '.' | count '=' | http | https | www | digit | letters | dir | use of ip | shorturls | | benign(wt) | 0.5385 | 0.4375 | 0.1666 | 0.5832 | 0.3334 | 0 | 0 | 1 | 0.4706 | 0.7272 | 0.8 | 0.1428 | 0.5 | 0.4 | 0 | 0 | | malicious(wt) | 0.46154 | 0.5625 | 0.83334 | 0.4167 | 0.6667 | 0 | 1 | 0 | 0.5294 | 0.2727 | 0.2 | 0.8572 | 0.5 | 0.6 | 1 | 0 |   With the help of these weights we will find rank of the urls’s , following is the table after multiplying weight by the quantity of the feature   |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | ID | Host | Path | Fdlen | Tldlen | count ‘\_' | count '@' | count '.' | | 1 | 0 | 15.8 | 0 | -0.4168 | 0 | 0 | 3 | | 2 | 0 | 14.06 | 0 | -0.4168 | 0 | 0 | 3 | | 3 | 6.02 | 7.87 | 0.8334 | 1.25 | 0 | 0 | 2 | | 4 | 5.1 | 3.37 | 4.167 | -0.4168 | 0 | 0 | 3 | | 5 | 0 | 12.94 | 0 | -0.4168 | 0 | 0 | 2 | | 6 | 13.44 | 0.5624 | 0 | 1.25 | 0.6667 | 0 | 2 | | 7 | 9.73 | 14.06 | 2.5 | 0.8336 | 0 | 0 | 6 | | 8 | 9.27 | 25.3 | 86.67 | 0.8336 | 0.667 | 0 | 2 | | 9 | 5.56 | 3.94 | 4.167 | 2.5 | 0 | 0 | 2 | | 10 | 7.5 | 0 | 0 | 1.75 | 0 | 0 | 0 | | 11 | 0 | 14.87 | 0 | -0.5832 | 0.33 | 0 | 0 | | 12 | 6.97 | 0 | 0 | 1.75 | 0 | 0 | 0 | | 13 | 8.04 | 0.4376 | 0 | 1.74 | 0 | 0 | 0 | | 14 | 0 | 7.001 | 0 | -0.5832 | 0 | 0 | 0 | | 15 | 9.65 | 7 | 0.4998 | -0.5832 | 0 | 0 | 0 | | 16 | 9.11 | 0.4376 | 0 | 1.75 | 0 | 0 | 0 | | 17 | 9.11 | 0.4376 | 0 | 1.75 | 0 | 0 | 0 | | 18 | 9.11 | 0.4376 | 0 | 1.75 | 0 | 0 | 0 | | | | | | | |
|  | | |  |  |  |  |
|  | | |  |  |  |  |
|  | | | |  |  |  |
|  | |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  | | | |  |  |  |
|  | | |  |  |  |  |
|  | | |  |  |  |  |
|  |  |  |  |  |  |  |
|  | | | |  |  |  |
| |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | count '=' | count http | count https | count www | count digit | count letter | count dir | use of ip | short urls | | 0 | 0.5294 | 0 | 0 | 12.83 | 4 | 1.799 | -1 | 0 | | 0 | 0.5294 | 0 | 0 | 6.85 | 4.5 | 1.799 | -1 | 0 | | 0 | 0.5294 | 0 | 0 | 0 | 13 | 1.799 | 1 | 0 | | 0 | 0.5294 | 0 | 0 | 6.85 | 4.5 | 0.5998 | -1 | 0 | | 0 | 0.5294 | 0 | 0.1998 | 0 | 7.5 | 1.799 | 1 | 0 | | 0 | 0.5294 | 0.2725 | 0 | 2.57 | 14 | 0.5998 | 1 | 0 | | 0 | 0.5294 | 0 | 0 | 6.85 | 16 | 2.39 | 1 | 0 | | 0 | 0.5294 | 0.2725 | 1.998 | 1.71 | 28 | 3.59 | 1 | 0 | | 0 | 0.4706 | 0.2725 | 0 | 0 | 10 | 1.199 | 1 | 0 | | 0 | 0.4706 | 0.7275 | 0.8 | 0 | 0 | 0 | 0 | 0 | | 0 | 0.4706 | 0.7275 | 0.8 | 0.1427 | 0 | 1.2 | 0 | 0 | | 0 | 0.4706 | 0.7275 | 0.8 | 0 | 0 | 0 | 0 | 0 | | 0 | 0.4706 | 0.7275 | 0.8 | 0 | 0 | 0.4002 | 0 | 0 | | 1 | 0.4706 | 0.0275 | 0.8 | 0 | 0 | 0 | 0 | 0 | | 0 | 0.4706 | 0.7275 | 0.8 | 0 | 0 | 1.2 | 0 | 0 | | 0 | 0.4706 | 0.7275 | 0.8 | 0 | 0 | 0.4002 | 0 | 0 | | 0 | 0.4706 | 0.7275 | 0.8 | 0 | 0 | 0.4002 | 0 | 0 | | 0 | 0.4706 | 0.7275 | 0.8 | 0 | 0 | 0.4002 | 0 | 0 | | | | | | | |
|  | | | | | | |
|  |  |  |  |  |  |  |
|  | | |  |  |  |  |
|  | | |  |  |  |  |
|  | | |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| ID | RANKING | RESULT |
| 1 | 2.15 | 1 |
| 2 | 1.81875 | 1 |
| 3 | 2.0575 | 1 |
| 4 | 1.65875 | 1 |
| 5 | 1.59125 | 1 |
| 6 | 2.305 | 1 |
| 7 | 3.74 | 1 |
| 8 | 5 | 1 |
| 9 | 1.945 | 1 |
| 10 | 0.7 | 0 |
| 11 | 1.12 | 0 |
| 12 | 0.669 | 0 |
| 13 | 0.8259 | 0 |
| 14 | 0.5259 | 0 |
| 15 | 1.235 | 0 |
| 16 | 0.856 | 0 |
| 17 | 0.8556 | 0 |
| 18 | 0.8556 | 0 |

SET THE THRESHOLD 1.5 HENCE THE RANKING OF ALL THE MALICIOUS MALE IS ABOVE 1.5 AND RANKING OF ALL THE BENIGN IS LESS THEN 1.5

# CONCLUSION: -

**The result we got by logistic regression model tells us the urls we got are malicious or benign and with help of statistical technique we are able to find the score of our logistic modal**